

DOCUMENT RESUME

ED 479 605

IR 022 209

AUTHOR Ronchetti, Marco; Rizzi, Matteo
TITLE Why Web Pages Annotation Tools Are Not Killer Applications? A New Approach to an Old Problem.
PUB DATE 2002-10-00
NOTE 6p.; In: E-Learn 2002 World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education. Proceedings (7th, Montreal, Quebec, Canada, October 15-19, 2002); see IR 022 093.
AVAILABLE FROM Association for the Advancement of Computing in Education (AACE), P.O. Box 3728, Norfolk, VA 23514. Tel: 757-623-7588; e-mail: info@aaace.org; Web site: <http://www.aace.org/DL/>.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE EDRS Price MF01 Plus Postage. PC Not Available from EDRS
DESCRIPTORS Computer Assisted Instruction; Computer Software Development; Cooperative Learning; Distance Education; Educational Technology; Instructional Materials; *Notetaking; *Web Sites; *World Wide Web

ABSTRACT

The idea of annotating Web pages is not a new one: early proposals date back to 1994. A tool providing the ability to add notes to a Web page, and to share the notes with other users seems to be particularly well suited to an e-learning environment. Although several tools already provide such possibility, they are not widely popular. This paper examines the reasons for this fact and proposes a novel architectural approach that is particularly well-suited to e-learning. Also presented is a tool that implements the proposed architecture. (Contains 14 references.) (Author/AEF)

C.H. Marks

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)

Why web pages annotation tools are not killer applications? A new approach to an old problem.

Marco Ronchetti and Matteo Rizzi

[marco.ronchetti, matteo.rizzi]@dit.unitn.it

Dipartimento di Informatica e Telecomunicazioni, Università di Trento
38050 Povo (T N) - Italy

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

This document has been reproduced as
received from the person or organization
originating it.

Minor changes have been made to
improve reproduction quality.

Points of view or opinions stated in this
document do not necessarily represent
official OERI position or policy.

Abstract

The annotation of web pages, and the possibility to share the notes with others, seems to be a powerful tool for an e-learning environment. Although several tools already provide such possibility, they are not widely popular, as one would expect them to be. We try to explain this fact, and propose a novel architectural approach that is, in our view, particularly suited to e-learning. We also present a tool that implements the proposed architecture.

Introduction

The idea of annotating Web pages is not a new one: early proposals date back to 1994, i.e. more or less to the pioneer-age of the WWW. Such functionality is appealing because it stresses the interactive character of the Web (that is most cases is instead sadly becoming more and more like a TV, where the highest level of "interaction" is to zap to another channel). Many uses can be supported by this functionality, so it is rather surprising that the idea did not have a large degree of success.

In particular, a tool providing the ability to add notes to a web page, and to share the note with other users seems to be particularly well suited to an e-learning environment. In fact, e-learning platforms aim not only at providing content to the student: in order to be effective they must support the learning process. For this reason virtually every e-learning suite provides at least some tool for asynchronous collaboration, like bulletin boards etc. Very often however these tool offer a poor degree of integration with the content of the e-courses. In many cases discussion among the "actors" (students and teachers) is carried on a forum: the forum though is a different tool, so it is separated from the presentation material. While sometimes this is perfectly acceptable, in many cases it is rather inconvenient since the discussion happens out of context, and many external references are necessary, making the whole process rather cumbersome.

Annotations on a Web page could in many cases be used instead of forums. A forum is certainly very well suited for the discussions of broad topics (as the metaphor implicit in the name implies). In many cases however the emphasis is on clarifying a point in a lecture. For instance, in the context of a programming course students are likely to discussing the code presented in an electronic lecture: in those circumstances, continuous references to "line X of code Y on page Z" make the writing and reading of the notes an exhausting task. Some discussions might even start using notes, and then move to forums. In other cases a discussion on a forum might be supported by notes included on a web page: we therefore see an interesting potential in using annotations.

Other uses might include assignments to the students that are published though a web site: in such a case the teacher could grade the assignment adding notes to it.

Yet another use could be a collaborative work made by students who cooperate producing a report: notes could be used in the same way, as revisions are possible using tools like MS Word.

Given the richness of scenarios that could be supported by web annotation, it is really puzzling that the tools publicly available on the web do not encounter more favor. In fact, since the early web years, sever projects have produced various tools for annotating pages. A list of them can be found on the "Collaboration, Knowledge Representation and Automatability" page of the W3C [COL], and a comparison among several such tools is available in [GAR].

We believe that lack of a wide success can be probably traced to a few different factors. On the one hand, too much a broad scope for an idea can be counterproductive: it is often better to address specific needs in a precise way rather than offering general support for a rather vague functionality. Most tools try to solve the problem of "annotating whatever page you want, wherever it is", and, to our knowledge, do not try to address a specific domain.

BEST COPY AVAILABLE

Another factor might be the absence of a ripe technology, that forces to use hacks ending up with a rather clumsy product. Early efforts were certainly penalized by the absence of a suitable technology. The resulting (annotated) pages were rather heavy, for instance using frames for showing the notes, or to add the annotations as footnotes, making the reading of the page sub-optimal. Recent attempts (e.g. [KOI]) try to leverage the newest technologies, like the Resource Description Framework (the RDF specifications [RDF] provide a lightweight ontology system to support the exchange of knowledge on the Web), and use the notion of Xlink [XLI] and Xpointer [XPO]. Such a road, although very interesting and probably correct on the long run, makes however the deployment of the tool now (and in the next few years) almost impossible: the only browser that supports such choices today is, to our knowledge, the experimental browser of W3C: Amaya [AMA]. In the light of these considerations, we revisit the idea of web pages annotation. We restrict the focus to the domain of e-learning; we examine the existing software architectures supporting the annotations, and propose a different architectural approach. We also examine the technology that is available *now* in a *standard* way to obtain a user interface that is suitable for our application domain. We also set as a requirement the possibility of integrating our proposed approach in (virtually all) e-learning platforms. Finally, as a proof of concept, we implement a portable tool that allows introducing annotations in existing e-learning systems.

Software architecture: possible alternatives

The annotation tools that are available today are based essentially on two alternative architectures. The first possibility is to modify the behavior of the browser, so after fetching a regular URL and rendering the related document, it allows writing notes that can be stored locally, or on a separated web server. Similarly, for every page it can enquiry the (local or remote) notes repository, and visualize existing notes with the document. The approach is certainly interesting, since it completely de-couples the original material from the notes. The choice of having the notes stored locally is limited, since it does not allow to interact with other people. The main limit of this approach is that modifying a browser is not a trivial task: the best solution could be one based on the notion of plug-in. However this would require maintaining different plug-ins for different browsers, and even for different implementations of the same browser (e.g. Windows vs. Mac vs. Linux, or version 5 vs. version 6).

The second major architecture is based on the concept of an "Annotation-server". The idea is to have an intermediate web server (proxy) that gets the requests, forwards them to the page server, gets the response and decorates it with the notes. Similarly, the proxy accepts the new notes and stores them. This architecture has some major advantages: it does not need a customized browser, and still preserves the separation between notes and content. However, the requested URL must be coded in a special way: "http://url_of_the_proxy?//http://url_of_the_resource": "real" URL is passed as a parameter to the proxy server. This can be an advantage (the same URL can be annotated through several proxies unaware of each other) but is also rather annoying, since every URL has to be "preprocessed" by the user in order to obtain the desired behavior. Moreover, following a link from a page does not pass through the annotation proxy, unless the proxy itself does not take the rather heavy burden of parsing all pages it delivers in order to suitably change all referenced URLs.

A third approach (not found in literature) could be possible: the Annotation-server could be registered by the user as the standard proxy used by the browser. In such way all requests pass through the Annotation server, and the problem of explicitly coding the URL would be completely solved. This solution is simple and brilliant, but it collides with the typical notion of a Proxy (a *local* machine, typically used for optimizing the network traffic, and often employed also for by-passing a local firewall).

By examining these architectures, one finds that the common denominator is the separation between the content server and the annotation server. This choice has obvious advantages:

- a) the content server needs not to be aware of the existence of the annotation server;
- b) one can define multiple annotation servers, supplying notes to (possibly closed) groups
- c) it is possible to annotate just about every resource available on the web.

However, the limited success of the existing system witnesses that the balance problems overcome the advantages: some companies that offered an annotations as a commercial service discontinued their offering [THI].

We therefore think that it is worth revising the fundamental choices. In particular, we try to drop the nice but ambitious idea to be able to annotate any possible page. We focus on an e-learning platform that typically gets the (main) content from a limited number of servers (very often just one): our goal becomes therefore to be able to annotate only the pages of these servers. At this point, even the separation between annotation server and content server becomes less relevant, and we can concentrate

the whole job on the same server. This approach does not require a customized browser neither needs to pass through an intermediate server, relieving both problems encountered above. The last problem to be solved is to make the look and feel of the pages appealing, and to integrate the two functions (providing content and notes) in the same server.

The solution we suggest is to add the annotation functionality to a web server as Java servlets [SER]. Servlets are platform independent, and are supported by virtually every Web Server available on the market. The servlet gets the request for a page, locally searches a database that stores the notes, and builds an annotated page that is served back to the user. Similarly, another servlet gets new notes and stored them in the database. The original page is almost unchanged: we only require that:

- a) It is written in valid XHTML (an HTML extension that is XML compliant)
- b) The page contains a definition of the contexts to which notes can be appended.

The transformation from normal (even incorrect!) HTML to XHTML can be easily done by a free tool (Dave Raggett's HTML TIDY [TYD]). The definition of the contexts can be done automatically, with the chosen granularity: for instance each phrase, or each paragraph could be a context).

The notes are rendered in place using the standard subset of JavaScript that is commonly recognized by the most diffused recent browsers (Netscape Navigator and Internet Explorer versions 4 and above, Opera, Mozilla).

In summary, the architecture we provide contemplates:

- a) an (automatic) conversion of HTML pages to a suitable (still HTML!) format
- b) the filtering of each request through a system based on servlets
- c) the storage of the notes on a database residing on the server that provides content.
- d) The rendering of the notes in various way based on JavaScript.

In addition, we introduce the notion of users and groups so that the visibility of the notes can be restricted to particular actors (a class, the teachers, on particular person etc.). Such addition reflects the spirit of supporting or particular domain of election, i.e. e-learning.

Implementation: WebNotes

To proof the concept, we implemented a tool called WebNotes according to the architecture we suggest.

A first art of WebNotes is an automatic tool that performs the preprocessing of the page. During the conversion process the granularity of the markable context that can be annotated is chosen: it can be each word, each phrase or each paragraph. As we mentioned the process is necessary in order to guarantee that the resulting pages are XHTML compliant, and that the contexts are suitably defined. The resulting page is still normal HTML, although it conforms the more restrictive rules that make it XML compliant.

The pages requested by the user are filtered through a servlet enabled Web server, which maintains a set of notes and their properties (like ownership, visibility etc). Notes are shown in the text as tiny icons embedded in the page. Public notes are visible to everybody. After the user identifies him/herself through a login process, s/he can see all notes allowed by her/his privileges.

The user can toggle between two modes: browsing and editing. While browsing, when s/he moves the mouse pointer over an icon that shows the presence of a note, the note pops up on place. When moving the mouse away the note disappears, unless it was stuck by the reader, in which case it disappears only when explicitly closed.

In order to edit the page (adding or modifying a note) the user enables editing mode and simply clicks on the location where s/he wants to add the note. A popup window appears, and the users can edit/modify the note. In the basic version the note contains only text, without any formatting. In an enhanced version of WebNotes, the window contains an applet that allows writing HTML in a simple "What-You-See-Is-What-You-Get" fashion. We kept the applet separated in order to minimize the requirements we make on the browser, and in order to avoid delays related to its downloading.

The system is currently in the final stages of development, and already underwent some extensive testing with different browsers.

Although WebNotes can be used in various contexts, it was explicitly designed for deployment in an e-learning environment. We envision its use for performing some of the discussions that presently happen on electronic forums.

At the moment of writing, we plan to use its preliminary version on the field in some e-courses at the University of Trento during the last bimester. By the time of the conference we expect to have gathered statistics on its use and feedback by the users.

Since WebNotes is made by a series of Java Servlets, it can be deployed on any Servlet-enhanced Web Server (nowadays virtually every serious Web Server can has a servlet-compliant engine). We do not assume any special requirement on the client side, the only request being the support of (the standard subset of) JavaScript (that is available on all recent browsers). We tested our annotated pages with all the major players (i.e. Opera, Mozilla, Netscape, Internet Explorer) on several platforms without encountering problems. The notes are simple text, so their writing and the editing is performed in a normal web page. Also the administration of the system (e.g. group creation and maintenance) occurs in a web page. An enriched version of our tool also includes a Java Applet that is a HTML composer: this allows formatting the notes in a reach way, using the full power of HTML. Since the Applet adds functionality but is not needed to make the system usable, WebNotes can be configured according to the preference of the deployer. Thanks to these choices WebNotes can be integrated in most e-learning platforms, enriching them with a new functionality.

Related Work

As we mentioned at the beginning, although much emphasis has been given to the support of collaboration via web since the early WWW years [COL], a convincing technology has not yet emerged. Interesting approaches run from an almost uncontrolled interactive editing of web pages (wiki's [CUN]) to more controlled annotations. In particular, adding notes to a Web page is not a new idea *per se*. Tools comparable to what we present have been proposed in the past, but in most cases their architecture was dissimilar from the one we've chosen. The main architectural choices were either based on browser-specific tools, or on external mediator servers. A comparison of some available tools can be found in [GAR].

Browser specific tools (e.g. [COM]) require the use of a customized browser, which is certainly an inconvenient choice.

In the proxy-based approach, the annotation process is taken care of by an intermediate server, that collects the notes, gets the original page and present to the user a page decorated it with the annotations. This allows taking control of the annotation process, and in some cases is offered by commercial enterprises [ANN] [THI]. A non-commercial instance of this class of tools is CritLink [CRI].

Our tool is rather based on an automatic pre-processing of an HTML page to include the fields that (with the chosen granularity) define the markable locations, and rely then on a basic browser and on a standard web-server architecture to deliver the content. This promises to be a winning point, since it allows an easier integration with existing e-learning platforms.

Bibliography

- [ANN] <http://www.annotate.net/> Annotate
- [AMA] Amaya, W3C's Editor/Browser, <http://www.w3.org/Amaya/>
- [COL] "Collaboration, Knowledge Representation and Automatability"
<http://www.w3.org/Collaboration/>
- [CUN] Cunningham, W.: wiki (wiki wiki web) <http://c2.com/cgi/wiki?WikiHistory>
- [CRI] CritLink <http://crit.org/http://crit.org/index.html>
- [COM] ComMentor <http://hci.stanford.edu/commentor/>, by Martin Röscheisen, Christian Mogensen, and Terry Winograd at the Stanford Digital Library Project
- [KOI] Koivunen, M.R. et al., "The W3C Collaborative Web Annotation Project ... or how to have fun while building an RDF infrastructure", submitted to UIST 2000,
<http://www.w3.org/2000/02/collaboration/annotation/papers/annotationinfrastructure>
- [GAR] Garfunkel, J., "Web Annotation Technologies",
<http://look.boston.ma.us/garf/webdev/annotate/software.html>
- [RDF] "Resource Description Framework", <http://www.w3.org/RDF/>
- [SER] "Java™ Servlet Technology" <http://java.sun.com/products/servlet/>

[THI] ThirdVoice <http://www.thirdvoice.com/> Following the link, one discovers that: "As of April 2, 2001, we are no longer supporting the consumer version of the our search and annotation tools."

[TID] Tidy <http://www.w3.org/People/Raggett/tidy/>

[XLI] XML Linking Language (XLink) Version 1.0, W3C Recommendation 27 June 2001, <http://www.w3.org/TR/xlink/>

[XPO] XML Pointer Language (XPointer) Version 1.0, W3C Candidate Recommendation 11 September 2001, <http://www.w3.org/TR/xptr/>



U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)



NOTICE

Reproduction Basis

X

This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐

This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").